

## Jによる花のグラフィックスー1 ーチューリップー

西川 利男

### 0. はじめに

先の JAPLA 研究会で志村正人氏より「花のCG」戸川隼人著、サイエンス社(1988)なる本を渡され、Jでプログラムが出来ないものかと依頼を受けた。

この本には、チューリップ、朝顔など約30種類の花の絵が、コンピュータグラフィックで描かれている。かつ、ていねいな説明とともにN88BASICのプログラムリストも付けられているので、簡単にと引き受けたが、仲々てこずった。しかし、夏の合宿として、かっこうの楽しいテーマであり、私自身興味もあったので、いくつかの花の絵をJのグラフィックスとしてプログラムとともに紹介する。

### 1. 「花のCG」ー N88BASICとJのプログラム

同書では一次関数、二次関数、指数関数、三角関数などの組み合わせで、花の花弁や葉、茎などを描き、色づけしている。

処理のメカニズムを見るために、N88BASICのプログラムは一応は動かしたが、その考え方を参考するものの、逐語的な変換はおよそ無理であり、J独自の方式でプログラムを作成することにした。

現在ではN88BASICを走らせる環境は極めて貴重であり、幸い手元にNEC PC9801VXが健在であり、テストすることができた。なお、インターネットで検索すると互換のWindows版のActiveBasicというのがあるようだが、試してはいない。

図形の表示位置計算のためのBASICのFOR ~ NEXTループはJでは得意とする配列処理でコンパクトに行なうことが出来る。

また、図形処理に関して

BASICの PSET, LINE, PAINT などは

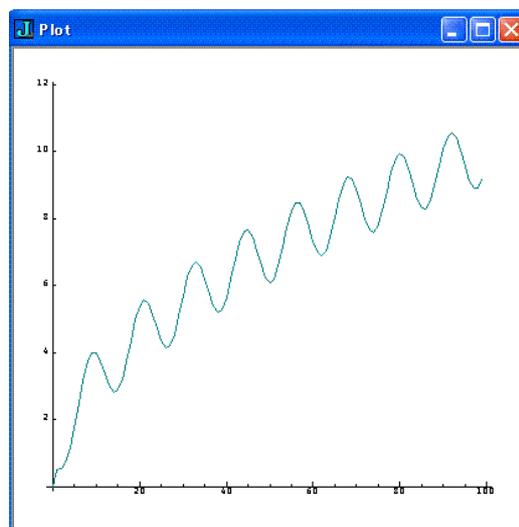
Jでは gllines, glrgb, glpolygon など  
gl2グラフィックス命令で対応した。

### 2. 花一輪を描く

簡単に花の輪郭となる点の位置座標を決めて、その内部を色で塗りつぶせばよい。

花卉や葉の曲線は次のように数学関数を利用して簡単に描くことが出来る。Jでちょっとやってみよう。

```
load 'trig plot'  
X =: i. 100  
Y1 =: X^0.5  
Y2 =: sin 100*X  
plot Y1 + Y2
```



## 2. 1 花卉

チューリップの花弁は次のような数学関数を用いて描くことができる。

外側の花弁  $f(t) = c*t^{0.1} + a*\sin b*x$

内側の花弁  $g(t) = c*t^1 + a*\sin b*x$

ここで、係数パラメータ  $a, b, c$  は絵を見ながら適宜きめる。

Jでは次のプログラムにより、赤い右半分の花弁が位置(XC, YC)に描かれる。

```
glrgb 255 0 0
glbrush ''
RX =: 70 [ RY =: 200 [ DR =: 0.1
YPP =: YC+H*RY*T
XAP =: XC+W*RX*((T^0.1)+DR*(sin(5.7*T)))
XBP =: XC+W*RX*((T^1)+DR*(sin(5.7*T)))
PA =: ,XAP,.YPP NB. Outer Points
PB =: ,XBP,.YPP NB. Inner Points
glpolygon PA, |. , |. "(1) 100 2$PB
```

これと対称的に左半分を合わせ、さらに中央部の花弁を追加して出来上がる。

## 2. 2 茎

茎は簡単に細い短冊形の図形で表した。

## 2. 3 葉

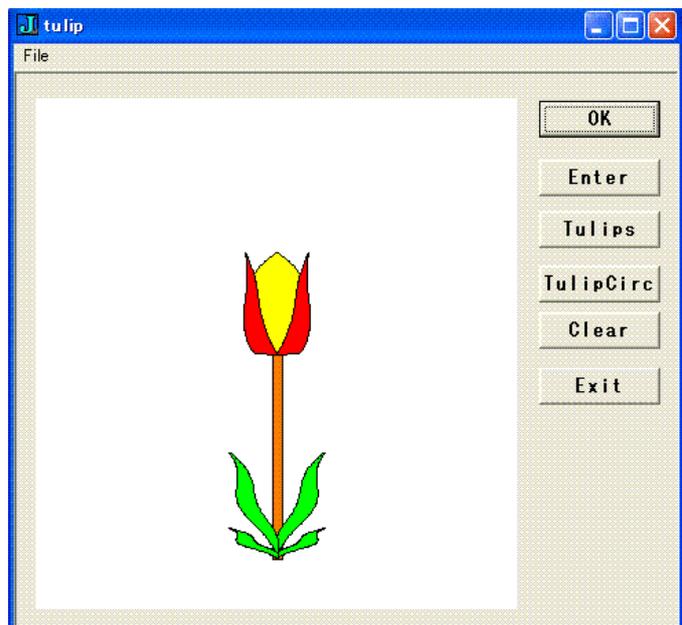
葉は少し変えた関数を使って、緑色で表した。

```
glrgb 0 255 0 NB. green
glbrush ''
YB =: YC+H*YD
YLV =: YB+H*210*T
XLA =: XC+W*100*(T+(0.2*sin(PI*T))-0.1*sin(3*PI*T))
XLB =: XC+W*100*(T+(_0.2*sin(PI*T))-0.1*sin(3*PI*T))
VA =: ,XLA,.YLV
VB =: ,XLB,.YLV
glpolygon VA, |. , |. "(1) 100 2$VB
```

このようにして、チューリップの花一輪が右のように描かれる。

チューリップの位置、サイズを指定するボタン[Enter]を設けた。

なお、Jのプログラムの詳細は最後のリストを参照のこと。

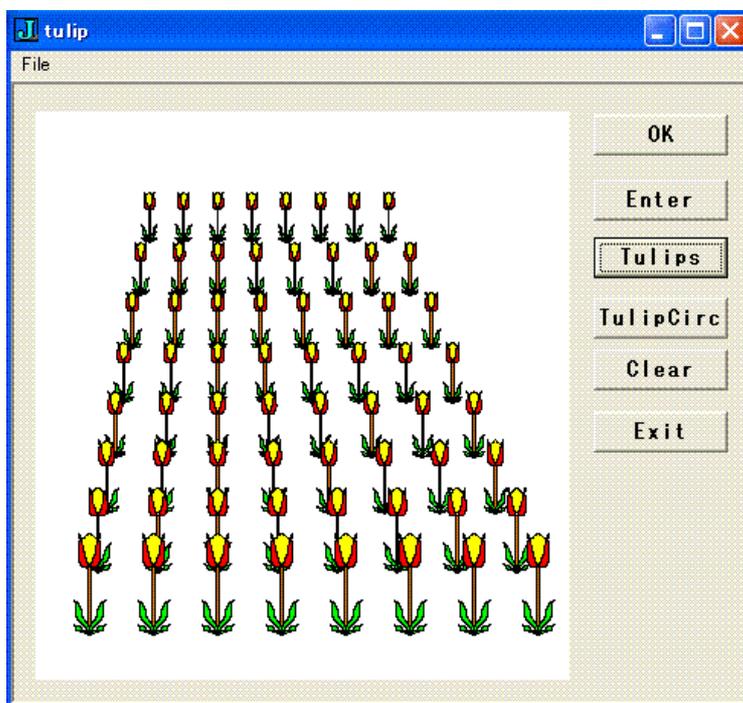


### 3. チューリップ畑

上のチューリップをたてよこに配置すれば、チューリップ畑ができる。

ここでは、一輪のローカル座標を画面全体のグローバル座標に変換する処理が必要になるが、Jの関数 `tulip_display` を位置を決めて行なえば造作はない。

ボタン `[tulips]` により実行される。

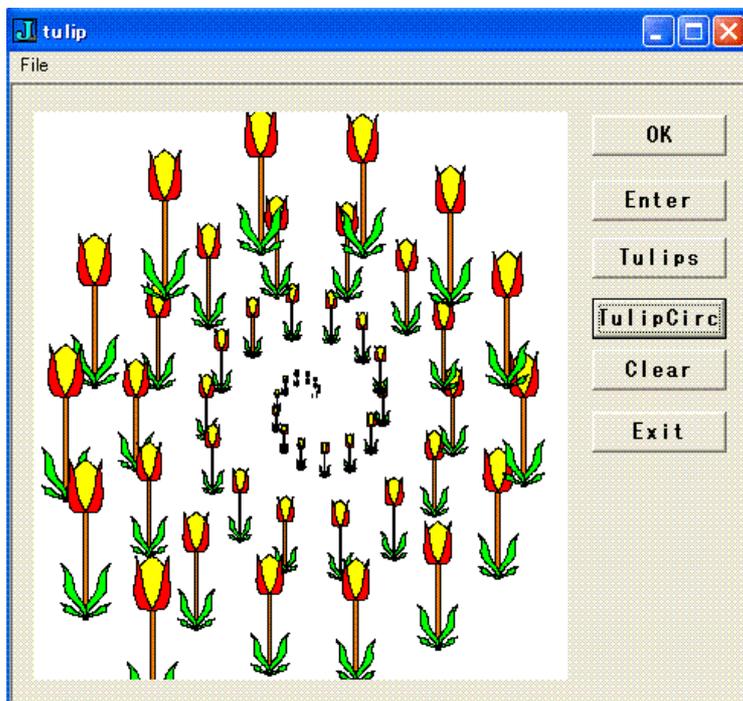


### 4. チューリップ円形花壇

チューリップをらせん状に配置すると、円形花壇となる。

アルキメデスのらせん  
 $r = a \theta$   
に沿って、サイズも変えて配置すれば、右のようになる。

ボタン `[TulipCirc]` により実行される。



NB. Tulip Graphics  
 NB. tulip.ijs  
 NB. from BASIC Programs, 戸川隼人「花のCG」サイエンス社、(1988).  
 NB. programmed by T. Nishikawa  
 NB. 2009/7/1 Tulip Flower  
 NB. 2009/7/3 Position, Size varied by Enter Button  
 NB. 2009/7/5 Tulips Arranged in Field, Tulips Spiral

```

require 'trig'
require 'gl2'

```

NB. base form

```

TULIP=: 0 : 0
pc tulip closeok;
menupop "File";
menu new "&New" "" "" "";
menu open "&Open" "" "" "";
menusep ;
menu exit "&Exit" "" "" "";
menupopz;
xywh 190 9 44 12;cc ok button;cn "OK";
xywh 190 96 44 12;cc cancel button;cn "Exit";
xywh 7 8 175 167;cc tulipgraf isigraph;
xywh 190 28 44 12;cc Enter button;
xywh 190 78 44 12;cc Clear button;
xywh 190 63 44 12;cc TulipCirc button;
xywh 190 45 44 12;cc Tulips button;
pas 6 6;pcenter;
rem form end;
)

```

```

run =: tulip_run
tulip_run=: 3 : 0
wd TULIP
NB. initialize form here
wd 'pshow;'
)

```

```

tulip_cancel_button=: 3 : 0
wd 'pclose;'
)

```

```

tulip_Clear_button=: 3 : 0
glclear ''

```

```

)
tulip_Enter_button=: 3 : 0
wd 'pc editbox;'
wd 'xywh 8 11 80 8;cc s0 static;cn "Enter X, Y, W, H?";'
wd 'xywh 80 10 100 10;cc e0 edit ws_border;'
wd 'setfocus e0;'
wd 'pas 8 8;pcenter;pshow;wait;'
wd 'pclose;'
DA =. wd 'q;'
'XC YC W H' =: ". , >((<'e0') = 0{"(1) DA) # (i.#DA) { 1{"(1) DA
)

```

```

tulip_ok_button=: 3 : 0
tulip_display XC, YC, W, H
)

```

NB. Tulip Display \*\*\*\*\*

```

PI =: 3.1416
T =: (i.100)%100 NB. Plot 100 Points
XC =: 500 NB. Root of Flower
YC =: 500 NB. Root of Flower
YD =: _400 NB. Root of Stem & Leaves
W =: 1
H =: 1
tulip_display=: 3 : 0
'XC YC W H' =. y.
RX =: 70 NB. petal parameter
RY =: 200 NB. petal parameter
DR =: 0.1

```

NB. tulip center petal =====

```

glrgb 255 255 0 NB. yellow
glbrush ''
YQA =: 25{. H*RY*(1-T)
YQZ =: ({:YQA), 5
YQQ =: YQA, YQZ
XQA =: 25{. W*50*sin(5.7*T)
XQZ =: ({:XQA), ({:XQA)
XQQ =: XQA, XQZ
QA =: , (XC+XQQ),. (YC+YQQ)
QB =: , (XC-XQQ),. (YC+YQQ)
QAA =: 27 2$QA
QBB =: |."(2) 27 2$QB
QCC =: QAA, QBB
glpolygon ,QCC

```

```

)

NB. tulip side petal =====
glrgb 255 0 0          NB. red
glbrush ''
NB. tulip petal data

NB. right petal - Plus Part =====
YPP =: YC+H*RY*T
XAP =: XC+W*RX*((T^0.1)+DR*(sin(5.7*T)))
XBP =: XC+W*RX*((T^1)+DR*(sin(5.7*T)))
PA =: ,XAP,.YPP NB. Outer
PB =: ,XBP,.YPP NB. Inner
glpolygon PA, |. , |. "(1) 100 2$PB
NB. gllines PA, |. , |. "(1) 100 2$PB
NB. left petal - Minus Part =====
XAP =: XC-W*RX*((T^0.1)+DR*(sin(5.7*T)))
XBP =: XC-W*RX*((T^1)+DR*(sin(5.7*T)))
PA =: ,XAP,.YPP NB. Outer
PB =: ,XBP,.YPP NB. Inner
glpolygon PA, |. , |. "(1) 100 2$PB
NB. gllines PA, |. , |. "(1) 100 2$PB

NB. tulip stem =====
glrgb 250 125 0        NB. brown
glbrush ''
NB. tulip stem data
ST =: (XC+W*_10), YC, (XC+W*10), YC
ST =: ST, (XC+W*10), (YC+H*YD)
ST =: ST, (XC+W*_10), (YC+H*YD)
ST =: ST, (XC+W*_10), (YC)
glpolygon ST

NB. tulip leaf =====
glrgb 0 255 0          NB. green
glbrush ''
YB =: YC+H*YD
NB. tulip leaf data
YLW =: YB+H*60*T
YLV =: YB+H*210*T
NB. Plus Part =====
XLA =: XC+W*100*(T+(0.2*sin(PI*T))-0.1*sin(3*PI*T))
XLB =: XC+W*100*(T+(_0.2*sin(PI*T))-0.1*sin(3*PI*T))
VA =: ,XLA,.YLV
VB =: ,XLB,.YLV

```

```

glpolygon VA, |. , |. "(1) 100 2$VB
WA =: ,XLA,.YLV
WB =: ,XLB,.YLV
glpolygon WA, |. , |. "(1) 100 2$WB
NB. Minus Part =====
XLA =: XC-W*100*(T+(0.2*sin(PI*T))-0.1*sin(3*PI*T))
XLB =: XC-W*100*(T+(_0.2*sin(PI*T))-0.1*sin(3*PI*T))
VA =: ,XLA,.YLV
VB =: ,XLB,.YLV
glpolygon VA, |. , |. "(1) 100 2$VB
WA =: ,XLA,.YLV
WB =: ,XLB,.YLV
glpolygon WA, |. , |. "(1) 100 2$WB
glshow ''
)

```

NB. Tulip Field

```

NB. Tulips Arranged at Position XP, YP and Size SZ(W, H)
tulip_Tulips_button=: 3 : 0
U =: i.8
V =: i.8
X0 =: 100 + (120*U)
XP =: X0 -"(1 1) 8 * V */ (_2+U)
Y0 =: 200 + (90*V)
YP =: (#U)#Y0
XYP =: (,XP),.YP
SZ =. 0.3 * % %: 0.005*YP
SZ2 =. SZ,.SZ
tulip_display L:0 <"(1) |. XYP, ("1) SZ2
)

```

NB. Tulip Spiral Display

```

TH =: PI * i.52
RS =: 100*TH
XS =: 520 + 0.03*RS*cos TH%7
YS =: 500 + 0.03*RS*sin TH%7
tulip_TulipCirc_button=: 3 : 0
XYS =: (XS,.YS)
SZS =: 0.3 * 0.01*TH
tulip_display (L:0) <"(1) YYS, ("1) SZS,.SZS
glshow ''
)

```

## g12 グラフィックスの J の版による仕様の違い —簡単なテストプログラムを作って—

西川 利男

### 1. J のバージョンとグラフィックス

J のシステムはウィンドウズ版にしても、J3, J4, J5 そして最新では J6 とバージョンアップを重ねてきた。

J6 になっての言語仕様の変更として、関数の引数が x. から x へ、y. から y へなどあるが、J6 の新しいプリミティブ M. (Memorize) について先に報告した[1]。

J6 のグラフィックスについても、わずかだが仕様の変更が行われているようである。筆者が J4 など以前作ったプログラムが J6 ではうまく動かないとのクレームが寄せられ、これを明らかにするため、簡単なテストプログラムを作って比較検討してみた。

ところで、J のグラフィックスには次のようにいろいろな方法がある。

- ① wd や g12 コマンドによるプログラミング
- ② plot ツール
- ③ viewmat ツール

まず、②と③とは簡便でかつ高性能であり、これについては志村正人氏の詳細な解説がある[2]。しかし、筆者はやや特殊なグラフィックスを行うことが多く、そのため①の基本的なコマンドでプログラミングを行わざるをえない[3]。

### 2. J のバージョンによる g12 コマンドの扱い方の違い

J のグラフィックスでは、例えば線を引くといった操作は gline などのコマンドによりおこなうが、この仕様が J のバージョンにより異なる。

J3 まで …… グラフィックス操作は wd (=windows driver) の文字列引数として行う

J4, J5 …… グラフィックスは専用の g12 コマンド群によりおこなう

J6 …… g12 コマンドの仕様の一部が廃止や変更された

ここで g12 コマンドとは J のプリミティブ 11!:2000+n に名前を付けたものである。

さらに、J の各バージョンによる基本の相違点を比較するとつぎのようになる。

	J402	J504	J602
クラスファイル jg12	不要	必要 coinsert 'jg12'	必要 coinsert 'jg12'
グラフの表示	glshow''	glshow''	glpaint''
表示の座標値	左下(0, 0) —右上(1000, 1000)	左下(0, 0) —右上(1000, 1000)	左上(0, 0) —右下(200, 200)

#### 文献

[1] 西川利男「J602 の新しい機能 M. について」JAPLA シンポジウム資料 2007/12/8

[2] 志村正人「J のグラフィックス (基礎編—1) (J6 版)」

JAPLA 研究会資料 2007/2/21

[3] 西川利男「J グラフィックスによる一般のレムニスケート図形」

JAPLA 研究会資料 2008/2/23

NB. Tulip Graphics  
 NB. Tulip.ijs for J602  
 NB. from BASIC Programs, : 책 lu CGvTCGX ĘA(1988).  
 NB. programmed by T. Nishikawa  
 NB. 2009/7/1 Tulip Flower  
 NB. 2009/7/3 Position, Size varied by Enter Button  
 NB. 2009/7/5 Tulips Arranged in Field, Tulips Spiral  
 NB. 2009/7/24 adjust for J602  
 require 'trig'  
 require 'gl2'  
 coinsert 'jgl2' NB. adjusted for J602

NB. base form

```

TULIP=: 0 : 0
pc tulip closeok;
menupop "File";
menu new "&New" "" "" "";
menu open "&Open" "" "" "";
menusep;
menu exit "&Exit" "" "" "";
menupopz;
xywh 191 9 34 12;cc ok button;cn "OK";
xywh 190 147 34 12;cc cancel button;cn "Exit";
xywh 7 8 175 167;cc tulipgraf isigraph;
xywh 190 28 34 12;cc Enter button;
xywh 190 126 34 12;cc Clear button;
xywh 190 63 34 12;cc TulipCirc button;
xywh 190 45 34 12;cc Tulips button;
xywh 189 99 37 12;cc Test button;cn "Test";
pas 6 6;pcenter;
rem form end;
)

```

```

run =: tulip_run
tulip_run=: 3 : 0
wd TULIP
NB. initialize form here
wd 'pshow;'
)

```

```

tulip_cancel_button=: 3 : 0
wd 'pclose;'
)

```

```

tulip_Clear_button=: 3 : 0
glclear ''
glpaint '' NB. adjust for J602
glpen 1, 0 NB. for J602
)

```

```

tulip_Enter_button=: 3 : 0
wd 'pc editbox;'
wd 'xywh 8 11 70 8;cc s0 static;cn "Enter X, Y, W, H?";'
wd 'xywh 70 10 80 10;cc e0 edit ws_border;'
wd 'setfocus e0;'
wd 'pas 8 8;pcenter;pshow;wait;'
wd 'pclose;'
DA =. wd 'q;'
'XC YC W H' =: ". , >((<'e0') = 0{"(1) DA) # (i.#DA) { 1{"(1) DA
)

```

```

tulip_ok_button=: 3 : 0
tulip_display XC, YC, W, H
NB. tulip_display 200, 200, 0.4, _0.4 NB. adjust for J602
)

```

NB. Tulip Display \*\*\*\*\*

```

PI =: 3.1416
T =: (i.100)%100 NB. Plot 100 Points
NB. XC =: 500 NB. Root of Flower
NB. YC =: 500 NB. Root of Flower
YD =: _400 NB. Root of Stem & Leaves
XC =: 200 NB. Root of Flower NB. for J602
YC =: 200 NB. Root of Flower NB. for J602

```

```

NB. W =: 0.2
NB. H =: 0.2
W =: 0.4 NB. for J602
H =: _0.4 NB. for J602
tulip_display=: 3 : 0
'XC YC W H' =. y NB. adjust for J602
RX =: 70 NB. petal parameter for J602
RY =: 200 NB. petal parameter for J602
DR =: 0.1
NB. RX =: 14 NB. petal parameter
NB. RY =: 200 NB. petal parameter
NB. DR =: 0.1

```

```

NB. tulip center petal =====
glrgb 255 255 0          NB. yellow
glbrush ''
YQA =: 25{. H*RY*(1-T)
YQZ =: ({:YQA), 5
YQQ =: YQA, YQZ
XQA =: 25{. W*50*sin(5.7*T)
XQZ =: ({:XQA), ({:XQA)
XQQ =: XQA, XQZ
QA =: , (XC+XQQ),. (_8+YC+YQQ) NB. adjust for J602
QB =: , (XC-XQQ),. (_8+YC+YQQ) NB. adjust for J602
QAA =: 27 2$QA
QBB =: |."(2) 27 2$QB
QCC =: QAA, QBB
glpolygon ,QCC

```

```

NB. tulip side petal =====
glrgb 255 0 0          NB. red
glbrush ''

```

```

NB. tulip petal data
NB. right petal - Plus Part =====
YPP =: YC+H*RY*T
XAP =: XC+W*RX*((T^0.1)+DR*(sin(5.7*T)))
XBP =: XC+W*RX*((T^1)+DR*(sin(5.7*T)))
PA =: ,XAP,.YPP NB. Outer
PB =: ,XBP,.YPP NB. Inner
glpolygon PA, |. , |."(1) 100 2$PB
NB. gllines PA, |. , |."(1) 100 2$PB
NB. left petal - Minus Part =====
XAP =: XC-W*RX*((T^0.1)+DR*(sin(5.7*T)))
XBP =: XC-W*RX*((T^1)+DR*(sin(5.7*T)))
PA =: ,XAP,.YPP NB. Outer
PB =: ,XBP,.YPP NB. Inner
glpolygon PA, |. , |."(1) 100 2$PB
NB. gllines PA, |. , |."(1) 100 2$PB

```

```

NB. tulip stem =====
glrgb 250 125 0          NB. brown
glbrush ''
NB. tulip stem data
ST =: (XC+W*_10), YC, (XC+W*10), YC
ST =: ST, (XC+W*10), (YC+H*YD)
ST =: ST, (XC+W*_10), (YC+H*YD)

```

```

ST =: ST, (XC+W*_10), (YC)
glpolygon ST

NB. tulip leaf =====
glrgb 0 255 0          NB. green
glbrush ''
YB =: YC+H*YD
NB. tulip leaf data
YLW =: YB+H*60*T
YLV =: YB+H*210*T
NB. Plus Part =====
XLA =: XC+W*100*(T+(0.2*sin(PI*T))-0.1*sin(3*PI*T))
XLB =: XC+W*100*(T+(_0.2*sin(PI*T))-0.1*sin(3*PI*T))
VA =: ,XLA,.YLV
VB =: ,XLB,.YLV
glpolygon VA, |. , |. "(1) 100 2$VB
WA =: ,XLA,.YLW
WB =: ,XLB,.YLW
glpolygon WA, |. , |. "(1) 100 2$WB
NB. Minus Part =====
XLA =: XC-W*100*(T+(0.2*sin(PI*T))-0.1*sin(3*PI*T))
XLB =: XC-W*100*(T+(_0.2*sin(PI*T))-0.1*sin(3*PI*T))
VA =: ,XLA,.YLV
VB =: ,XLB,.YLV
glpolygon VA, |. , |. "(1) 100 2$VB
WA =: ,XLA,.YLW
WB =: ,XLB,.YLW
glpolygon WA, |. , |. "(1) 100 2$WB
glpaint '' NB. adjust for J602
)

NB. Tulips Position XP, YP and Size SZ(W, H)

tulip_Tulips_button=: 3 : 0
U =: i.8
V =: i.4
NB. X0 =: 100 + (120*U)
X0 =: 20 + (60*U) NB. for J602
XP =: X0 -"(1 1) 8 * V */ (_2+U)
NB. Y0 =: 200 + (90*V)
Y0 =: 300 - (80*V) NB. for J602
YP =: (#U)#Y0
XYP =: (,XP),.YP
NB. glrgb 255 0 0
NB. glbrush ''

```

```

NB. glellipse (L:0) <"(1) |. XYP, ("1) 10, 10
NB. glpaint ''
SZ =. 0.2 * 0.3 * % %: 0.005*YP NB. for J602
SZ2 =: SZ,.-SZ NB. adjust for J602
tulip_display L:0 <"(1) XYP, ("1) |. SZ2
)

```

```

NB. Tulip Spiral Display
tulip_TulipCirc_button=: 3 : 0
TH =: PI * i.28
RS =: 100*TH
XS =: 180 + 0.02*RS*cos TH%7 NB. for J602
YS =: 160 - 0.02*RS*sin TH%7 NB. for J602
XYS =: (XS,.YS)
glrgb 255 0 0
glbrush ''
NB. glellipse (L:0) <"(1) |. YYS, ("1) 10, 10
NB. glpaint '' NB. adjust for J602
SZS =: 0.3 * 0.01*TH
tulip_display (L:0) <"(1) YYS, ("1) SZS,.-SZS
)

```

```

tulip_Test_button=: 3 : 0
NB. glines 500 500 500 700 700 700 700 500 500 500
glrgb 0 255 0
glbrush ''
NB. glflood 600 600 0 255 0
NB. glellipse 100 100 50 50
NB. glellipse 200 100 50 50
NB. glellipse (L:0) 100 100 50 50;200 100 50 50
NB. XYDA =: <"(1) XY, ("1) 50, 50
glellipse (L:0) <"(1) |. XYP, ("1) 10, 10
NB. glellipse (L:0) <"(1) |. YYS, ("1) 50, 50
glpaint '' NB. adjust for J602
)

```